Extremal throughputs in free-choice nets

Anne Bouillard[1], Bruno Gaujal[1] and Jean Mairesse[2]

[1] Lip (Umr Cnrs, Ens Lyon, Inria, Université Claude Bernard Lyon 1),
École Normale Supérieure de Lyon,
46, allée d'Italie - 69364 Lyon Cedex 07 - France
{abouilla,bgaujal}@ens-lyon.fr
[2] Liafa, Cnrs - Université Paris 7
Case 7014 - 2, place Jussieu - 75251 Paris Cedex 5 - France
mairesse@liafa.jussieu.fr

Abstract. We give a method to compute the throughput in a timed live and bounded free-choice Petri net under a total allocation (*i.e.* a 0-1 routing). We also characterize and compute the conflict-solving policies that achieve the smallest throughput in the special case of a 1-bounded net. They do not correspond to total allocations, but still have a small period.

1 Introduction

Petri nets are logical objects, originally and above all. However, the interest of Petri nets for modelling purposes has induced the need for timed and stochastic extensions of the model. Performance evaluation then becomes a central issue, and the throughput is arguably the main performance indicator.

Consider now a live and bounded free-choice Petri net (LBFC). Such Petri nets realize a good compromise between modelling power and mathematical tractability, see [6] for several striking examples of the latter. Assume that the Petri net is timed with a timing specified by a constant real-valued firing time for each transition. To remove the undeterminism of the behavior of the Petri net, a policy for the resolution of all the conflicts needs to be decided. Once it is chosen, all the enabled transitions start to fire as soon as possible, and the time that elapses between the beginning and the completion of the firing of a transition is equal to the firing time. Therefore, the timed evolution of the Petri net is completely determined.

Our goal is to study the global *activity* or *throughput* or *firing rate of the transitions* of the Petri net in a sense to be made precise later on. Of course, the activity depends on the chosen policy for resolving conflicts.

In a free-choice Petri net, one may view a conflict-solving policy as a set of local functions associated with conflict places, and assigning tokens to output transitions. The simplest class of policies consists of the so-called *0-1 policies*: for a conflict place p, allocate all the tokens to a fixed transition. Zero-one policies are called *total allocations* in [6]. The next simplest class of policies is, arguably, the *periodic* ones: for a conflict place p, allocate the tokens to the output transitions according to some fixed periodic pattern. Obviously, 0-1 policies are also periodic policies.

In this paper, we address the following natural questions:

– A. Given a periodic policy, is the activity explicitly computable?
– B. Consider the set of all possible, arbitrarily complex, policies for resolving conflicts. Is the infimum, resp. supremum, of the activity over this set attained by a 0-1 or a periodic policy? Can we explicitly determine the policies realizing the infimum, resp. supremum?

For both questions, we are also concerned by the algorithmic complexity of the computations.

Consider first Question A. It is known that the activity is explicitly computable when the timings are rational-valued [3]. The solution relies on the construction of a very large finite graph G in which a state incorporates three different types of information: the current marking; the remaining time before completion for the currently firing transitions; and the current position of the cursor within the periods for the periodic policy. The timed behavior is ultimately periodic and the period corresponds to an elementary circuit in the graph G. The activity is computed along this circuit.

The method has two major drawbacks. First, it is not efficient from an algorithmic point of view. Indeed, the graph G is in general much larger than the reachability (marking) graph whose size may already be exponential in the one of the Petri net. Second, it does not provide much insight on the structure of the timed behavior.

Here, we show that both restrictions can be overcome in the special case of a 0-1 policy: the live part of the Petri net becomes a disjoint union of event graphs. Consequently, the activity can be computed in polynomial (cubic) time in the size of the Petri net, using classical results on the throughput of timed event graphs [1, 4, 5]. Furthermore, the previous restriction on having rational timings is not necessary anymore for 0-1 policies.

Consider now Question B and assume that the timings are rational-valued for simplicity. Using a simplified version of the above graph G, in which the periodic policy is not coded anymore, one can easily prove that the supremum and the infimum of the activity are obtained for periodic policies. The drawbacks are the same as before: the time complexity, and the lack of structural insight. Concerning the latter, the method does not allow to answer the question: is the supremum or infimum attained by 0-1 policies?

Presumably against the intuition, we exhibit an example of a live and 1-bounded Petri net with firing times all equal to one, and for which the infimum is attained only by non-0-1 periodic policies. More generally, we show that for 1-bounded Petri nets with general $(0, \infty)$-valued timings, the infimum is attained by a periodic policy which may not be 0-1 but which can be characterized at the net level and which has a very small period (*i.e.* bounded by the total number of tokens). The same result fails to hold for a k-bounded Petri net, $k \geq 2$ where the general structure of infimum policies is not understood. An example is given of a 2-bounded net with timings all equal to one, for which the infimum is attained by periodic policies which are not 0-1 nor have small periods. To be complete, let us mention that the general structure of supremum policies is not well understood, even for 1-bounded nets. It is easy to build examples of 1-bounded LBFC with rational timings for which the supremum is attained only by non-0-1 Sturmian-like periodic policies, as well as examples with irrational timings for which the supremum is attained only by Sturmian-like non-periodic policies, see [9] and Example 6.2.

In order to obtain the above results, we use three different types of building blocks:

- The theory of timed event graphs;
- A structural result stating that the live part of a LBFC with a total allocation is a disjoint union of T-components; and that, given a T-component, there exists a total allocation making this T-component the only live part of the LBFC;
- The notion of Token-Transition invariants. It is a refinement of the classical notion of T-invariants with a dynamical flavor to it, since it "follows" the evolution of a token.

The first point is very classical [1, 4, 5], while the other two may be original and of some interest by their own.

The paper is organized as follows. The known results on Question A appear in Section 3. In Section 4, we study the 0-1 policies in detail. Section 5 introduces the TT-invariants. Section 6 is devoted to Question B. In particular, we characterize the policies which provide the infimum throughput for a 1-bounded net in Subsection 6.3.

Due to lack of space, all the proofs are not included. The missing proofs can be found in the corresponding research report [2].

2 Notations and preliminaries

Denote by $\mathbb{N}$ the nonnegative integers, and by $\mathbb{N}^*$ the positive integers. Given a set T and a subset S, denote by $\chi_S : T \to \{0,1\}$ the characteristic function of S in T defined by: $\chi_S(u) = 1$ if $u \in S$ and $\chi_S(u) = 0$ if $u \in T \backslash S$.

A *net* is a bipartite directed graph $(\mathcal{P}, \mathcal{T}, \mathcal{F})$ with $\mathcal{P} \cup \mathcal{T}$ as the set of nodes ($\mathcal{P} \cap \mathcal{T} = \emptyset$) and $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ as the set of arcs. A *Petri net* is a quadruple $(\mathcal{P}, \mathcal{T}, \mathcal{F}, M)$, where $(\mathcal{P}, \mathcal{T}, \mathcal{F})$ is a net and M is a map from $\mathcal{P}$ to $\mathbb{N}$. We sometimes write the Petri net $\mathcal{N}$ as $(\mathcal{N}, M)$ to emphasize the special role of M. The elements of $\mathcal{P}$ are called *places* and are represented by circles and those of $\mathcal{T}$ are called *transitions* and represented by rectangles. The function M is called the *(initial) marking* of the net and is represented by tokens in places. Let

$x \in \mathcal{P} \cup \mathcal{T}$ be a node. We denote by ${}^\bullet x$ the set of its predecessors and by $x^\bullet$ the set of its successors. We also set ${}^\bullet X = \cup_{x \in X} {}^\bullet x$ and $X^\bullet = \cup_{x \in X} x^\bullet$. A transition is *conflicting* if one of its input place has at least two successors. Otherwise, the transition is *non-conflicting*.

The marking evolves according to the *firing rule*. A transition t is *enabled* if: $\forall p \in {}^\bullet t,\ M(p) \geq 1$. An enabled transition can *fire*, and then the marking becomes M' with $M'(p) = M(p) - \chi_{{}^\bullet t}(p) + \chi_{t^\bullet}(p)$.

If the marking M' is obtained from M by firing the transition t, we write $M \xrightarrow{t} M'$. If M' is obtained by successively firing $\sigma = t_1 t_2 \cdots t_n \in \mathcal{T}^*$, we write $M \xrightarrow{\sigma} M'$. The sequence σ is called ad *(admissible) firing sequence*. Finally, if M' can be reached from M by firing some sequence, we write $M \to M'$. The set of the *reachable markings* of M is $\mathcal{R}(\mathcal{G}, M) = \mathcal{R}(M) = \{M' \mid M \to M'\}$.

A Petri net is *live* if for every transition t and every reachable marking M_1 there exists a marking M_2, reachable from M_1, that enables t. A Petri net is *deadlock-free* if there exists no reachable marking in which no transition is enabled. A Petri net is *k-bounded*, $k \in \mathbb{N}$, if for every reachable marking, the number of tokens in a place is less or equal to k. A Petri net is bounded if it is k-bounded for some k. A net $\mathcal{N}$ is *structurally live* if there exists a marking M such that the Petri net $(\mathcal{N}, M)$ is live. A net $\mathcal{N}$ is *well-formed* if there is a marking M that makes the Petri net $(\mathcal{N}, M)$ live and bounded

An *event graph* is a (Petri) net where: $\forall p \in \mathcal{P},\ |{}^\bullet p| = |p^\bullet| = 1$. A *state machine* is a (Petri) net where: $\forall t \in \mathcal{T},\ |{}^\bullet t| = |t^\bullet| = 1$. A *free-choice (Petri) net* is a (Petri) net where: $\forall (p,t) \in \mathcal{P} \times \mathcal{T},\ (p,t) \in \mathcal{F} \Rightarrow (p^\bullet = \{t\})$ or $({}^\bullet t = \{p\})$. We use the acronym *LBFC* for a live and bounded free-choice Petri net. A *choice-free (Petri) net* is a (Petri) net where: $\forall p \in \mathcal{P},\ |p^\bullet| = 1$.

The *incidence matrix* of a Petri net is $N \in \mathbb{Z}^{\mathcal{P} \times \mathcal{T}}$ with $N_{p,t} = \chi_{t^\bullet}(p) - \chi_{{}^\bullet t}(p)$. Let $\sigma \in \mathcal{T}^*$ be a firing sequence. The *commutative image* (or *Parikh vector*) of σ is $\overrightarrow{\sigma} = (|\sigma|_t)_{t \in \mathcal{T}}$, the vector of the number of occurrences of each transition t in σ. If $M \xrightarrow{\sigma} M'$, then the equation $M' = M + N\overrightarrow{\sigma}$ is satisfied.

Invariants of Petri nets. A column vector $J \in \mathbb{N}^{\mathcal{T}} \backslash \{(0, \ldots, 0)^T\}$ (resp. $I \in \mathbb{N}^{\mathcal{P}} \backslash \{(0, \ldots, 0)^T\}$) is a *T-invariant* (resp. *S-invariant*) if $NJ = 0$ (resp. $I^T N = 0$). A T-invariant (resp. S-invariant) is *minimal* if it is minimal for the component-wise ordering among all the T-invariants (resp. S-invariants). A subnet $\mathcal{N}'$ of the net $\mathcal{N}$ with the set of nodes X is a *T-component* (resp. *S-component*) if for every transition t of X, ${}^\bullet t \cup t^\bullet \subseteq X$ (resp. for every place p of X, ${}^\bullet p \cup p^\bullet \subseteq X$) and $\mathcal{N}'$ is a strongly connected event graph (resp. state machine). If $(\mathcal{P}_1, \mathcal{T}_1, \mathcal{F}_1)$ is a T-component (resp. S-component) of the net $\mathcal{N}$, then $\chi_{\mathcal{T}_1}$ (resp. $\chi_{\mathcal{P}_1}$) is a minimal T-invariant (resp. S-invariant) of $\mathcal{N}$. For a well-formed free-choice net, the converse is true: if J is a minimal T-invariant (resp. S-invariant), set $\mathcal{T}_1 = \{t \in \mathcal{T} \mid J_t \neq 0\}$ (resp. $\mathcal{P}_1 = \{p \in \mathcal{P} \mid J_p \neq 0\}$) and $\mathcal{P}_1 = {}^\bullet \mathcal{T}_1 \cup \mathcal{T}_1^\bullet$ (resp. $\mathcal{T}_1 = {}^\bullet \mathcal{P}_1 \cup \mathcal{P}_1^\bullet$), then $(\mathcal{P}_1, \mathcal{T}_1, \mathcal{F}_1)$ is a T-component (resp. S-component). See for instance [6, Prop. 5.7, Prop. 5.14, Th. 5.17].

A set of T-components (resp. S-components) forms a *T-cover* (*S-cover*) if every node belongs to one of these components. Well-formed free-choice nets are covered by T-components and also by S-components ([6, Theorems 6.6 and 5.18]).

We will also need the following result.

Theorem 2.1. *[6, Theorem 5.9] Let p be a place of a live and bounded free-choice Petri net $(\mathcal{N}, M)$. The bound of p is* $\min\{\sum_{s\in\mathcal{P}_1} M(s) \mid p \in \mathcal{P}_1,\ (\mathcal{P}_1, \mathcal{T}_1, \mathcal{F}_1)$ *is a S-component of* $\mathcal{N}\}$.

Clusters. The *cluster* $[x]$ of $x \in \mathcal{P} \cup \mathcal{T}$ is the smallest subset of $\mathcal{P} \cup \mathcal{T}$ such that: (i) $x \in [x]$; (ii) $p \in \mathcal{P}, p \in [x] \Rightarrow p^\bullet \in [x]$; (iii) $t \in \mathcal{T}, t \in [x] \Rightarrow {}^\bullet t \in [x]$. The set of all the clusters of a net defines a partition of the nodes of the net. For free-choice nets, each cluster contains only one place or only one transition.

Blocking marking. Let $(\mathcal{N}, M)$ be a Petri net and t a non-conflicting transition of $\mathcal{N}$. A *blocking marking* of the transition t is a reachable marking such that the only enabled transition is t. If t is a conflicting transition, a *blocking marking* of t is a reachable marking such that the only enabled transitions belong to the cluster $[t]$.

In [8, Theorem 3.1], it is shown that in a connected LBFC, for any transition b, there exists a unique blocking marking M_b. Moreover, M_b is reachable from any other reachable marking without firing b.

Timed and routed nets. A timed Petri net is a Petri net in which timings have been added on places and transitions. With no loss of generality, we only consider timings on the transitions, and not on the places. We also consider non-null timings. This is assumed for convenience. The results of the paper could be generalized with null timings, under the assumption that it is not possible to have an infinite number of firings occurring in 0 time. Set $\mathbb{R}_+^* = (0, +\infty)$. A timed Petri net is denoted by $(\mathcal{N}, M, \tau)$ with $(\mathcal{N}, M)$ a Petri net and $\tau \in (\mathbb{R}_+^*)^{\mathcal{T}}$ the vector of the timings. The timed semantics is the following one. Consider a transition t with timing τ_t which gets enabled at instant d. If the transition t is fired, the firing occurs as follows:

- At time d, the firing begins. A token is frozen in each input place of t and cannot enable another transition.
- At time $d+\tau_t$, the firing ends. The frozen tokens are removed from the input places of t and one token is added in each output place of t.

Observe that it is possible for a transition to have several ongoing firings at a given instant. The resulting evolution is called *as soon as possible* (*asap*), since a firing transition begins to fire as soon as it is enabled.

Any conflict-solving policy may be viewed as a set of local *routing functions* at each conflicting place. The global routing function is a vector $u = (u_p)_{p\in\mathcal{P}}$ where u_p is a function from $\mathbb{N}^*$ to $p^\bullet$. The k-th token arriving in place p (we consider the tokens in place p in the initial marking as the first arriving tokens) can only enable the transition $u_p(k)$. So the notion of enabled transition is modified by the routing function. A transition can be fired if all its input places contain a token which is routed to that transition. We denote by $(\mathcal{N}, M, u)$ a routed Petri net with routing u and by $(\mathcal{N}, M, \tau, u)$ a timed and routed Petri net.

A marking is *reachable* for a routed Petri net $(\mathcal{N}, M, u)$ if it is reachable for $(\mathcal{N}, M)$ via a firing sequence compatible with u. The notions of boundedness and liveness of $(\mathcal{N}, M, u)$ are defined accordingly.

If the Petri net $(\mathcal{N}, M)$ is bounded, so is the routed Petri net $(\mathcal{N}, M, u)$. If the Petri net $(\mathcal{N}, M)$ is live, $(\mathcal{N}, M, u)$ is not necessarily live, nor deadlock-free. However, if $(\mathcal{N}, M)$ is a LBFC, the routed net $(\mathcal{N}, M, u)$ cannot have any deadlock, because choices and synchronizations are separated. Hence, if a routed free-choice net is not live, we can always define its non-empty live part.

A routing $u = (u_p)_{p\in\mathcal{P}}$ is *periodic* if u_p is a periodic function for every p. A routing u is *0-1* if: $\forall p \in \mathcal{P}$, u_p is a constant function. A 0-1 routing is called a *total allocation* in [6].

3 Throughput in routed free-choice Petri nets

With no loss of generality, all Petri nets considered are assumed to be connected.

Consider a timed Petri net and let $\sigma = \sigma(1)\sigma(2)\cdots \in \mathcal{T}^{\mathbb{N}}$ be an infinite firing sequence. Set $\sigma_n = \sigma(1)\cdots\sigma(n)$. Consider the timed evolution starting at instant 0 and associated with σ. The *activity* $A(\sigma)$ of σ is the asymptotic average number of firings per unit of time:

$$A(\sigma) = \liminf_{n\to\infty} \frac{n}{d(n)}$$

where $d(n)$ is the first instant of completion of all the firings from σ_n.

To make this definition more general and more flexible, it is possible to "weight" the activity of each transition.

A *weight* $\alpha_t \in \mathbb{R}_+$ is associated to each transition t, we set $\alpha = (\alpha_t)_{t\in\mathcal{T}}$, and we assume that $\alpha \neq (0, \ldots, 0)$. The *throughput* $D(\sigma)$ of σ (for the weight α) is defined by:

$$D(\sigma) = \liminf_{n\to\infty} \frac{\sum_{t\in\mathcal{T}} \alpha_t (\overrightarrow{\sigma_n})_t}{d(n)} .$$

If all the weights are equal to one, the throughput is equal to the activity. On the other hand, if $\alpha_t = 1, \alpha_{t'} = 0, t' \neq t$, then the throughput measures the firing rate of transition t.

The above notion of throughput allows one to modify a Petri net without changing its throughput, for example, by replacing a transition of timing n and weight α by n transitions of timing 1 and weight α/n.

3.1 Periodic routings

Consider a timed and routed LBFC $(\mathcal{N}, M, \tau, u)$ with a *periodic* routing and *integer* firing times (rational firing times can be treated in a similar way).

The state of the Petri net at time t is a triple (M_t, R_t, U_t) where M_t is the marking at time t, R_t is the remaining firing time of all the current firings at time t and U_t is the current routing decision in all the routing places. Observe that the number of states is finite and bounded by $(k+1)^{|\mathcal{P}|} \times F^{k|\mathcal{T}|} \times L^{|\mathcal{P}|}$, where k is a bound on the number of tokens per place, F is a bound on the firing times of all the transitions and L is a bound on the period of the routing at each place.

Since the behavior of the net is deterministic, the net jumps from one state to its unique successor at each time-step.

The state space being finite, there exists a state which is visited twice for the first time, and the whole behavior becomes periodic from that point on. This shows that the throughput exists and can be computed along the periodic behavior of the net. However, this computation may have a very high complexity (in time and in space) because the state space is potentially huge.

A construction similar to the above one is proposed in [3].

3.2 A particular case: event graphs

In a live and bounded event graph $(\mathcal{G}, M)$, there is no routing place, hence no routing. In that case, it is useless to sweep the whole state space to compute the throughput. It is well-known that the firing rate is the same for all transitions and the throughput is given by:

$$D = \frac{\sum_{t \in \mathcal{T}} \alpha_t}{\rho(\mathcal{G}, M)}, \quad \text{where} \quad \rho(\mathcal{G}, M) = \max_{c \text{ circuit of } \mathcal{G}} \frac{\sum_{t \in c} \tau_t}{\sum_{p \in c} M(p)}. \tag{1}$$

The throughput can be computed in cubic time using Karp's algorithm, see for instance [1]. The constant $\rho(\mathcal{G}, M)$ is usually called the *cycle time* of $(\mathcal{G}, M)$ (see [1, 4, 5] for details).

4 Zero-one policies

In this section, we consider 0-1 routing policies instead of arbitrary periodic routings. We show that all the combinatorial difficulties of periodic routings can be overcome for 0-1 routings.

4.1 Total allocations and 0-1 policies

An *allocation* is a function u from a set of clusters C to $\mathcal{T}$ such that: $\forall c \in C,\ u(c) \in c$. A transition is *allocated* if it belongs to the image of u. An allocation is *total* if it is defined on all clusters. An allocation *points to* C if for every place p not belonging to C, there exists a path π from p to a place of C such that every transition along the path π is allocated.

A firing sequence σ agrees with an allocation $u : C \to \mathcal{T}$ if it does not contain any transition t such that $[t] \in C$ and $t \neq u([t])$.

Lemma 4.1. *[6, Lemma 6.5] Let C be a set of clusters of a strongly connected free-choice Petri net $\mathcal{N}$ and let $\bar{C}$ be the complementary set of C in the clusters of $\mathcal{N}$. Then there exists an allocation u defined on $\bar{C}$ that points to C, and if M is a bounded marking and $M \xrightarrow{\sigma}$ is an infinite sequence that agrees with u, then some transition of C is fired an infinite number of times in σ.*

In this paper, we see total allocations as 0-1 routing policies in the places: each place routes all its tokens to its unique allocated output transition.

When u is a 0-1 routing, $(\mathcal{N}, u)$ is a free-choice Petri net where all the transitions which are not allocated can be removed. Therefore, exactly one output transition remains for each place. We obtain a choice-free Petri net. We first study general choice-free nets before giving a characterization of the choice-free nets obtained as the live part of a free-choice Petri net with a 0-1 routing.

4.2 Choice-free nets

A *siphon* is a set of places R such that ${}^{\bullet}R \subseteq R^{\bullet}$. A *trap* is a set of places R such that $R^{\bullet} \subseteq {}^{\bullet}R$.

Lemma 4.2. *Consider a strongly connected and live choice-free Petri net. It is an event graph if and only if it is bounded.*

A connected live and bounded Petri net is strongly connected. So it follows from the above lemma, that a connected and live choice-free Petri net is either unbounded, or bounded in which case it is a strongly connected event graph.

4.3 Live part of a LBFC with a 0-1 routing

Consider a LBFC $(\mathcal{N}, M)$ with a 0-1 routing. Let $\widetilde{\mathcal{N}}$ be the net obtained by removing all the transitions (together with their input and output arcs) which are not chosen by the 0-1 routing. This means that the net $\widetilde{\mathcal{N}}$ may not be strongly connected anymore (some places may have no inputs). Figure 4.3 shows the construction of $\widetilde{\mathcal{N}}$ on an example.

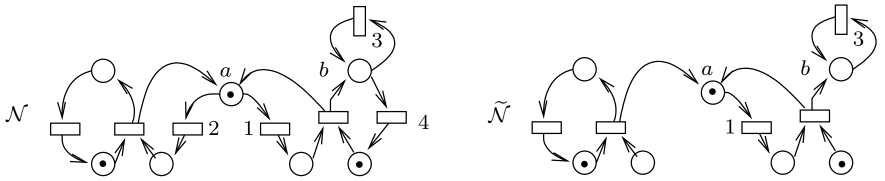

Fig. 1. The net $\widetilde{\mathcal{N}}$ is constructed by removing all non-allocated transitions. The 0-1 routing sends all tokens to transition 1 in routing place a and all tokens to transition 3 in routing place b. Transitions 2 and 4 are removed to construct $\widetilde{\mathcal{N}}$.

The Petri net $(\widetilde{\mathcal{N}}, M)$ is choice-free and bounded, on the other hand it may not be live. We are interested in characterizing the live part of $(\widetilde{\mathcal{N}}, M)$. This live part may depend on M. More precisely, let M' be a reachable marking of $(\mathcal{N}, M)$, let $(\mathcal{N}_1, M_1)$ be the live part of $(\widetilde{\mathcal{N}}, M)$ and let $(\mathcal{N}_2, M_2)$ be the live part of $(\widetilde{\mathcal{N}}, M')$. We may have $\mathcal{N}_1 \neq \mathcal{N}_2$, as well as $\mathcal{N}_1 = \mathcal{N}_2, \mathcal{R}(\mathcal{N}_1, M_1) \neq \mathcal{R}(\mathcal{N}_2, M_2)$.

The net $\widetilde{\mathcal{N}}$ can be decomposed into non-trivial maximal strongly connected components (mscc). There are two kinds of such components: the final components and the non-final components.

Lemma 4.3. *The final mscc are T-components of $\mathcal{N}$. The non-final mscc are event graphs, but not T-components of $\mathcal{N}$. The final mscc may or may not be live in $\widetilde{\mathcal{N}}$, the non-final mscc are not live in $\widetilde{\mathcal{N}}$.*

Lemma 4.4. *Let $(\mathcal{G}, M)$ be a live and 1-bounded event graph. Then*
- if one token is removed, the event graph is not live anymore,
- if one token is added, the event graph is not 1-bounded anymore.

As a consequence of Lemma 4.3 and Lemma 4.4, we have the following theorem.

Theorem 4.1. *Let $(\mathcal{N}, M_0)$ be a LBFC with a 0-1 routing. Let $\widetilde{\mathcal{N}}$ be the net obtained from $\mathcal{N}$ by removing the arcs and transitions which are not selected by the routing. The live part of $(\widetilde{\mathcal{N}}, M_0)$ is a non-empty disjoint union of strongly connected event graphs $(\mathcal{G}_1, M_1), \ldots, (\mathcal{G}_k, M_k)$, where each $\mathcal{G}_i$ is a T-component of $\mathcal{N}$. If $(\mathcal{N}, M_0)$ is 1-bounded, or if $k = 1$, then the sets of reachable markings $\mathcal{R}(\mathcal{G}_1, M_1), \ldots, \mathcal{R}(\mathcal{G}_k, M_k)$, do not depend on the 0-1 routing such that the live part of $(\mathcal{N}, M_0)$ consists of $\mathcal{G}_1, \ldots, \mathcal{G}_k$.*

If $\mathcal{G}$ is a T-component of $\mathcal{N}$, then there exists a 0-1 routing such that the live part of $(\widetilde{\mathcal{N}}, M_0)$ is precisely $(\mathcal{G}, M)$ for some marking M.

4.4 Throughput of 1-bounded free-choice nets with 0-1 routings

We now have all the ingredients to prove the main result.

Theorem 4.2. *Consider a LBFC $(\mathcal{N}, M_0)$ with a 0-1 routing. Assume that the live part is $(\mathcal{G}, M)$, where $\mathcal{G}$ is a single T-component corresponding to the T-invariant $J_{\mathcal{G}}$. Then the throughput does not depend on the 0-1 routing such that $\mathcal{G}$ is the live part, and it is equal to*

$$D_{\mathcal{G}} = \frac{\alpha^T J_{\mathcal{G}}}{\rho(\mathcal{G}, M)}.$$

Assume that the live part is $(\mathcal{G}_1, M_1), \cdots, (\mathcal{G}_k, M_k)$, where $\mathcal{G}_i$ is a T-component. Then the throughput is

$$D = \sum_{i=1}^{k} D_{\mathcal{G}_i} = \sum_{i=1}^{k} \frac{\alpha^T J_{\mathcal{G}_i}}{\rho(\mathcal{G}_i, M_i)}.$$

If the free-choice Petri net is 1-bounded, then this throughput does not depend on the 0-1 routing such that the live part consists of $\mathcal{G}_1, \ldots, \mathcal{G}_k$.

Proof. The proof easily follows from (1) and Theorem 4.1. Observe that the cycle time $\rho(\mathcal{G}, M)$ of an event graph depends on M only through the token count of circuits, see (1).

5 Token-Transition-Invariants

Let $(\mathcal{N}, M)$ be a Petri net. Let $\sigma = \sigma_1 \cdots \sigma_k$ be a firing sequence. We say that $\sigma' = \sigma'_1 \cdots \sigma'_h$, $h \leq k$ is a *subsequence* of σ if there exists an increasing function $f : \{1, \dots, h\} \to \{1, \dots, k\}$ such that $\sigma'_i = \sigma_{f(i)}$ for all $i \in \{1, \dots, h\}$.

Definition 5.1 (compatible firing sequence). *Let $\pi = p_1 t_1 p_2 t_2 \cdots p_\ell t_\ell$, $p_i \in \mathcal{P}$, $t_i \in \mathcal{T}$ be a path of $\mathcal{N}$ and M be a marking that marks p_1. Let σ be an admissible firing sequence of $(\mathcal{N}, M)$. The sequence σ is* compatible with π *if the first subsequence of σ in $[t_1][t_2] \cdots [t_\ell]$ is $t_1 t_2 \cdots t_\ell$.*

In the above definition, *first* has the following meaning. Order all subsequences of σ according to the point-wise ordering of the increasing functions used to defined them ($f \leq g$ is $f(i) \leq g(i)$ for all i in the domain of f). *First* means smallest according to this ordering.

In other words, an admissible firing sequence σ is compatible with a path in the Petri net if all the transitions along that path are fired and in that order when σ is fired. This means that the token which was initially in place p_1 successively enters places $p_2, \dots, p_\ell$ when σ is fired.

Definition 5.2 (Token-Transition-Invariant). *Let $c = p_1 t_1 p_2 t_2 \cdots p_\ell t_\ell$ be a circuit of $\mathcal{N}$ and M be a marking that marks p_1. A vector $J \in \mathbb{N}^{\mathcal{T}}$ is a Token-Transition-invariant (or TT-invariant) generated by c and the marking M if it is a T-invariant and if it is the commutative image of an admissible firing sequence compatible with c.*

A TT-invariant J generated by c and the marking M is minimal if for every other TT-invariant J' generated by c and the marking M, J' is not smaller than J.

A TT-invariant generated by c is minimal if for every other TT-invariant J' generated by c and some marking, J' is not smaller than J.

In words, a TT-invariant is a T-invariant such that one token has moved along a circuit and is back to its original place when the corresponding sequence is fired (hence the name).

In spite of what the definition suggests, TT-invariants generated by c do not depend on the initial marking: if the commutative image of $\sigma_1 \cdot \sigma_2$ is a TT-invariant for c, so is the commutative image of $\sigma_2 \cdot \sigma_1$, since it is a firing sequence from the marking M' such that $M \xrightarrow{\sigma_1} M'$.

However, unlike general T-invariants, TT-invariants depend on the set of reachable markings. We will see in the following that they actually mainly depend on the maximal possible number of tokens in circuit c.

The following Lemma characterizes minimal TT-invariants in event graphs, where things are easy.

Lemma 5.1. *Let c be an elementary circuit of a live and 1-bounded event graph containing n tokens. The minimal TT-invariant generated by c is $(n, \ldots, n) \in \mathbb{N}^{\mathcal{T}}$.*

Proof. Recall that the T-invariants of an event graph are of the form $(x, \ldots, x)$, $x \in \mathbb{N} \backslash \{0\}$, see [6, Prop. 3.16]. Now, let J be a minimal TT-invariant associated with $c = p_1 t_1 \cdots$, and let σ be a corresponding compatible firing sequence. Since the Petri net is 1-bounded, tokens along the circuit c cannot overtake each other. Hence, when the token initially in p_1 is back to p_1, after the firing of σ, we know that transition t_1 must have fired n times.

Example 5.1. Figure 2 shows the evolution of an event graph containing 2 tokens in circuit c. We look at the minimal TT-invariant generated by c. Using Lemma 5.1, the minimal TT-invariant is $(2, 2, 2, 2, 2, 2, 2, 2, 2)$. The white token (as well as the black one) is back to its original place (Figure 2(c)).

The minimal T-invariant is $(1, 1, 1, 1, 1, 1, 1, 1, 1)$. Note that, after a single firing of every transition (Figure 2(b)), the marking is unchanged , but the white token has switched its position with the black one. After firing every transition again (Figure 2(c)), the white token is back in the right place.

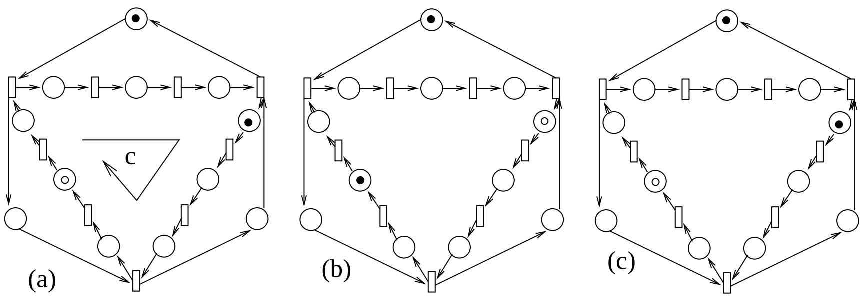

Fig. 2. TT-invariant in an event graph

We now characterize the minimal TT-invariants generated by a circuit of a live and 1-bounded free-choice Petri net. This is a more difficult case.

Let us first give an algorithm to build every minimal TT-invariant (Lemma 5.2 and Proposition 5.1). The following results (Lemma 5.3 and Proposition 5.2) show that a minimal TT-invariant generated by a circuit c is the sum of at most n minimal T-invariants, where n is the maximal number of tokens that c may contain.

Lemma 5.2. *Let c be a circuit of $\mathcal{N}$ a live and 1-bounded free-choice net, and b be a transition of c. Recall that M_b is the unique blocking marking associated with b. For every minimal TT-invariant J, there exists a firing sequence σ compatible with c such that $J = \overrightarrow{\sigma}$ that can be fired from M_b.*

Proposition 5.1. *Let c be a circuit of $\mathcal{N}$, a live and 1-bounded free-choice net. Every minimal TT-invariant generated by c is found by applying Algorithm 1.*

Algorithm 1 : Construction of minimal TT-invariants

Input: $\mathcal{N}$, a live and 1-bounded free-choice net, $c = t_1 \cdots t_k, t_{k+1} = t_1$ circuit of $\mathcal{N}$.
Output: A minimal TT-invariant generated by c.
$\sigma \leftarrow \varepsilon$;
for all $i \in \{1, \cdots, k\}$ **do**
  $\sigma_i \leftarrow$ a minimal firing sequence from M_{t_i} to $M_{t_{i+1}}$ in $\mathcal{N}$;
$\sigma \leftarrow \sigma_1 \ldots \sigma_k$;
Return $\overrightarrow{\sigma}$.

In the algorithm, a firing sequence $\sigma : M \xrightarrow{\sigma} M'$ is a *minimal* firing sequence if it does not contain any subsequence $\sigma' : M \xrightarrow{\sigma'} M'$. Such a minimal firing sequence has no reason to be unique. Hence the algorithm may yield several different outputs for a given input.

We now show that a minimal TT-invariant generated by an elementary circuit is the sum of n minimal T-invariants, where n is the maximal number of tokens in c (it is given by the number of tokens in c under the blocking marking of any transition of c). Lemma 5.3 is a lemma used to prove Proposition 5.2.

Lemma 5.3. *Let c be an elementary circuit of $\mathcal{N}$, a live and 1-bounded free-choice net and n be the maximal number of tokens in c. There exists a minimal TT-invariant generated by c that is n times the same minimal T-invariant.*

Proposition 5.2. *Let c be an elementary circuit of $\mathcal{N}$, a live and 1-bounded free-choice net and n be the maximal number of tokens in c. Every minimal TT-invariant generated by c is the sum of n minimal T-invariants.*

Example 5.2. Figure 3 illustrates Proposition 5.2: consider the circuit $c = 1, 2, 3, 5, 6, 7, 9, 10, 11$. The blocking marking of transition 1 is $\{d, g, k\}$. To reach the blocking marking of transition 2, the two possible firing sequences are $1, 9, 10, 11$ or $1, 9, 12$. To reach the blocking marking of transition 3, 2 is fired, and to reach the blocking marking of transition 5, 3 is fired. By symmetry of the net, there are also two possible minimal firing sequences from M_5 to M_6, that are $5, 1, 2, 3$ or $5, 1, 4$ and from M_9 to M_{10}, $9, 5, 6, 7$ or $9, 5, 8$. Every other minimal firing sequence from a blocking marking to the next on circuit c is made of only one transition. Then, there are 8 minimal TT-invariants (for three subsequences, there are two possibilities). For example, $1, 9, 10, 11, 2, 3, 5, 1, 4, 6, 7, 9, 5, 8, 10, 11$ is a firing sequence compatible with c whose commutative image is a minimal TT-invariant, which can be written as $I + J$, where I and J are minimal T-invariants, of respective support $1, 2, 3, 5, 6, 7, 9, 10, 11$ and $1, 4, 5, 8, 9, 10, 11$.

Lemma 5.4. *Let $\mathcal{C}$ be a circuit in a live and 1-bounded free-choice net and J be a minimal TT-invariant of $\mathcal{C}$. The circuit is composed of the elementary circuits*

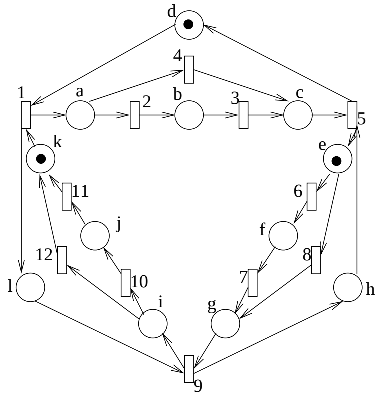

Fig. 3. Example of a free-choice net.

$c_1, \ldots, c_k$. Then there exists k minimal TT-invariants, $J_1, \ldots, J_k$ respectively generated by $c_1, \ldots, c_k$ such that $J = \sum_{i=1}^{k} J_i$.

6 Extremal Throughputs

We now consider all possible, arbitrarily complex, routing policies and try to address the following questions: Can one compute the routing policy which yields the best or worst throughput? Are the best and worst policies periodic? Are the best and worst policies 0-1?

6.1 Dominance of periodic policies

In this section we show that the best and worst throughputs in LBFC with rational firing times are achieved by periodic policies, and we provide an algorithm to construct them. The construction is very close to the one given in Section 3.1 and is basically the one in [3].

Consider a timed LBFC with non-null rational timings $(\tau_t)_{t \in \mathcal{T}}$ and with a vector of weights $\alpha = (\alpha_t)_{t \in \mathcal{T}}$, see Section 3. Let $x \in \mathbb{Q}$ be such that $\tau_t = k_t x, k_t \in \mathbb{N} \backslash \{0\}$, for all $t \in \mathcal{T}$. Build a new LBFC by replacing each transition by a path: a transition with timing τ_t and weight α_t is replaced by k_t transitions of timing 1 and weight $\alpha_t/(k_t x)$. Consider an infinite firing sequence σ for the original Petri net and the corresponding firing sequence σ' in the new Petri net. Then the throughputs, defined as in Section 3, associated with σ and σ' coincide. Therefore we can, without loss of generality, consider LBFC with timings all equal to 1.

Asap marking graph of a free-choice net. Consider a timed LBFC $(\mathcal{N}, M, \tau)$ and assume that all the timings are equal to 1: $\forall t \in \mathcal{T},\ \tau_t = 1$. Denote by N

the incidence matrix. In a given marking, a transition may be enabled several times. So the transitions that can be fired simultaneously form a multi-set.

The *asap marking graph* is defined as follows:

$Q \leftarrow \{M\}$; Arc $\leftarrow \emptyset$; $\tilde{Q} \leftarrow \{M\}$;
while $\tilde{Q} \neq \emptyset$ **do**
  Pick $M' \in \tilde{Q}$;
  for all maximal multi-set U of transitions that can be fired simultaneously from M' **do**
    $M'' \leftarrow M' + N.U$;
    if $M'' \notin Q$ **then**
      $Q \leftarrow Q \cup \{M''\}$; $\tilde{Q} \leftarrow \tilde{Q} \cup \{M''\}$;
    Arc $\leftarrow$ Arc $\cup \left\{M' \rightarrow M'', \text{ with label and weight } [U \mid \sum_{t \in U} \alpha_t]\right\}$;
  $\tilde{Q} \leftarrow \tilde{Q} \setminus \{M'\}$;

The above construction stops because the Petri net is bounded. So the *asap* marking graph is finite.

All the *as soon as possible* (*asap*) evolutions of the Petri net can be read on this graph, hence its name. In this graph, the *weight* of a path is the sum of the weights of the arcs. The *average weight* of a path is its weight divided by its length (number of arcs).

Theorem 6.1. *Let $(\mathcal{N}, M, \tau)$ be a timed LBFC with non-null rational timings. The minimal and maximal throughputs are obtained for periodic routings.*

Proof. Let $(\widetilde{\mathcal{N}}, \widetilde{M})$ be the LBFC obtained from $(\mathcal{N}, M)$ after duplicating the transitions such that every transition in the new Petri net has timing 1. Consider the *asap* marking graph of $(\widetilde{\mathcal{N}}, \widetilde{M})$. Let c be a circuit of the *asap* graph of maximum average weight. The maximal throughput can be reached by following this circuit.

Also, the minimal throughput can be reached by following the circuit of minimal average weight.

This shows that the corresponding routings are periodic. Indeed the routing can be deduced from the labels along the circuit of the *asap* marking graph. In particular, the period is smaller than the length of the circuit.

Example 6.1. Consider the Petri net of Figure 3. Every transition has timing 1 and weight 1. The *asap* marking graph is represented in Figure 4.

The minimal average weight of a circuit is 15/9, given by the circuit $\{(dek), (afl), (cgl), (chi), (dej), (dgk), (agl), (bhi), (chk)\}$. This gives the minimal throughput and the routing to reach it. The maximal average weight is attained for the circuit $\{(dek), (agl), (chi)\}$. And the maximal throughput is 2.

When the graph is built, computing the throughput can be made in cubic time in the number of nodes of the *asap* marking graph. But this graph can have an exponential size in the size of the original Petri net. One reason is that the number of markings of the net can be exponential in the size of the net. The other reason is that transitions are duplicated to build the Petri net with timings

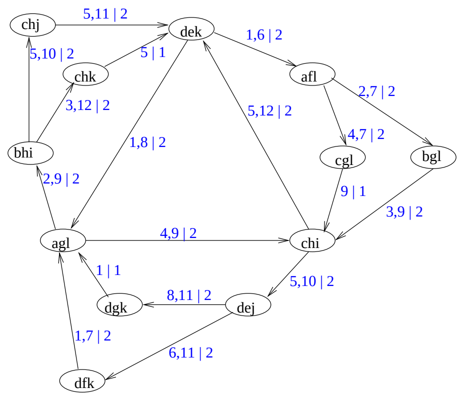

Fig. 4. Asap marking graph corresponding to the Petri net of Figure 3.

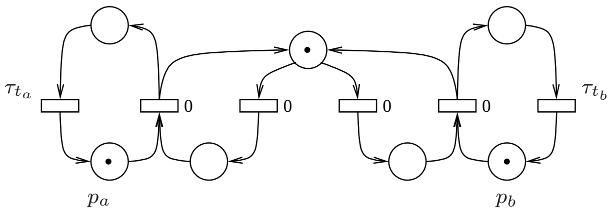

Fig. 5. One-bounded Petri net for which the optimal routing is not periodic.

1 starting from a Petri net with rational timings. Moreover, this method gives no information about the structure of the corresponding extremal policies.

6.2 Non-rational timings

For general non-rational timings, the maximal throughput is not always periodic, even for one-bounded nets, as shown in the following example.

Example 6.2. Look at Figure 5. The value of the firing times are given next to the transitions on the Figure. We have $\tau_{t_a}/\tau_{t_b} \notin \mathbb{Q}$. This model has been studied in [9]. The optimal routing is Sturmian aperiodic. The best routing consists in choosing the left (t_a) or right transition (t_b) depending on whether one token appears first in p_a or in p_b. If t_a has fired n_a times and t_b n_b times, it suffices to compare $n_a\tau_a$ and $n_b\tau_b$. The non-periodicity comes from the irrationality of the ratio of the timings.

The same Petri net can be considered with non-null rational timings approximating the ones in Figure 6.2. In this case, the maximal throughput is achieved for a Sturmian-like periodic routing policy. In particular, it is not possible to give an absolute bound for the period.

The above behaviors contrast sharply with the results to be proved in Section 6.3 on the minimal throughput for 1-bounded nets.

For general non-rational timings, we do not know if the minimal throughput is always attained by periodic policies. In the next section, we show however that it is the case for the subclass of 1-bounded Petri nets.

6.3 Minimal throughput in 1-bounded free-choice nets

Consider a timed live and 1-bounded free-choice net $(\mathcal{N}, M, \tau)$. In this part, we show that the minimal throughput is obtained for a periodic routing even for general non-rational timings. Furthermore, we give a precise insight on the structure of the periodic routing reaching the minimal throughput. Roughly it corresponds to a critical TT-invariant.

From Theorem 4.2, we can easily deduce the following lemma:

Lemma 6.1. *The worst 0-1 routing can be chosen among those that make only one T-component live.*

Critical circuit. Suppose again that the timings are 1. The worst routing can be read on the *asap* marking graph by considering an elementary circuit, $c = (M_0, \cdots, M_{T-1})$, of minimal average weight. The length of c is T, the period of the evolution.

Let t_0 be a live transition of the net appearing in the label between states M_0 and M_1 of the circuit c. We build a path in the Petri net with final extremity t_0 in the following recursive way.

If the path $t_i, \cdots, t_1, t_0$ is built, we choose the transition t_{i+1} in the label U_i of the arc between $M_{T-i-1[T]}$ and $M_{T-i[T]}$ and such that $t_{i+1} \in {}^{\bullet\bullet}t_i$.

We stop the construction when we find $m \in \mathbb{N}^*$ and a transition t_j such that $t_j = t_{j-mT}$. Consider the circuit of the Petri net corresponding to the sequence of transitions $t_j, t_{j-1}, \ldots, t_{j-mT+1}$, and denote it by $\mathcal{C}_c$. The length of this circuit is mT by construction.

Let U_i be the set of transitions whose firing leads from $M_{T-i-1[T]}$ to $M_{T-i[T]}$. (Here U_i is a set and not a multi-set, because the Petri net is 1-bounded.)

Let K be the commutative image of $\sigma = U_{j-1} \cdots U_{j-mT}$. By construction, K is a T-invariant. Also by construction, $(t_{j-1} \cdots t_{j-mT})$ is the first sub-word of σ belonging to $[t_{j-1}], \ldots, [t_{j-mT}]$. So, K is a TT-invariant generated by $\mathcal{C}_c$.

Since K is a T-invariant associated with a firing sequence following a minimal weight circuit in the *asap* marking graph, we deduce that the worst throughput $\underline{D}$ of $(\mathcal{N}, M, \tau)$ satisfies:

$$\underline{D} = \frac{\alpha^T K}{mT} .$$

Since K is a TT-invariant for $\mathcal{C}_c$, there exists J_c a minimal TT-invariant for $\mathcal{C}_c$ such that $K \geq J_c$. The circuit $\mathcal{C}_c$ may not be elementary. In full generality, it is composed of, say, k elementary circuits $c_1, \cdots, c_k$ of length $\ell_1, \cdots, \ell_k$, with $\sum_{i=1}^{k} \ell_i = mT$. By Lemma 5.4, we have $J_c = \sum_{i=1}^{k} J_i$ where J_i is a minimal TT-invariant of c_i. Then,

$$\underline{D} = \frac{\alpha^T K}{mT} \geq \frac{\alpha^T J_c}{mT} = \frac{\alpha^T (\sum_{i=1}^k J_i)}{\sum_{i=1}^k \ell_i} = \frac{\sum_{i=1}^k \alpha^T J_i}{\sum_{i=1}^k \ell_i} \geq \min_{i=1}^k \frac{\alpha^T J_i}{\ell_i} .$$

Now, by definition, $\underline{D} \leq \min_{i=1}^k (\alpha^T J_i)/\ell_i$. Therefore, we can find an elementary circuit c of the Petri net, of length ℓ and of associated minimal TT-invariant J, such that $\underline{D} = \alpha^T J/\ell$. Such a circuit is called a *critical circuit* of the Petri net.

We are now ready to state the main result of this section. Set $\mathbb{R}_+^* = (0, +\infty)$.

Theorem 6.2. *Consider a timed live and 1-bounded free-choice net with general timings in $\mathbb{R}_+^*$. Let α be a weight on the transitions. The minimal throughput is obtained for a periodic routing. For each place, the period of the routing function is bounded by the maximal number of tokens in the net.*

Example 6.3. Consider again the example of Figure 3. Form the corresponding asap marking graph shown on Figure 4, $(1, 2, 3, 5, 6, 7, 9, 10, 11)$ is the critical circuit. Since every transition has weight 1, the firing sequences of minimal weight from the blocking marking of a transition of c to the blocking marking of the next transition of c are (starting from the blocking marking of transition 1 which is $\{d, k, g\}$): $(1, 9, 12)$; (2); (3); $(5, 1, 4)$; (6); (7); $(9, 5, 8)$; (10); (11). The minimal TT-invariant achieving the minimal throughput is composed of two minimal T-invariant of support $1, 2, 3, 5, 6, 7, 9, 10, 11$ and $1, 4, 5, 8, 9, 12$. The minimal throughput is then $15/9$ as anounced before. As for the worst routing policy, it can be obtained directly from the minimal TT-invariant: $u_a = (2, 4)^\infty, u_e = (8, 6)^\infty$ and $u_i = (12, 10)^\infty$.

6.4 Algorithm to compute a routing that minimizes the throughput

Consider a timed live and 1-bounded free-choice net $(\mathcal{N}, M, \tau)$. Let N be the incidence matrix. Let Clusters be the set of clusters. Define the matrix $K \in \{0, 1\}^{\text{Clusters} \times \mathcal{T}}$ such that $K_{a,t} = 1$ if t belongs to the cluster a and $K_{a,t} = 0$ otherwise.

Let S be a subset of $\mathcal{T}$. Let **Lightest-T-invariant**(S) be the algorithm that computes a minimal T-invariant of minimal weight that contains the transitions of S. It is the solution of the following linear programming problem:

Algorithm 2 : Lightest-T-invariant

Input: $S \subseteq \mathcal{T}$
Minimize $\alpha^T \cdot I$
With constraints $N \cdot I = 0$; $K \cdot I \leq (1, \ldots, 1)^T$; $I \geq \chi_S$.

When the algorithm is called when S is a set of transitions belonging to the same T-component, the condition $K \cdot I \leq (1, \ldots, 1)^T$ ensures that the output is a set of disjoint minimal T-invariants, with only one containing S.

This algorithm runs in polynomial time since it is a linear program over $\mathbb{Q}$. Due to the form of the constraints, the solution will always be $\{0,1\}$-valued.

Let us define the functions **Blocking-Marking**(t), **Cycle-time**(A), and **Timing**(c).

– **Blocking-Marking**(t) computes the blocking marking of transition t. This marking can be computed in time $O(|\mathcal{T}|^2)$ when the Petri net is an event graph, and in time $O(|\mathcal{T}|^3)$ for the general free-choice case, see [8].
– **Cycle-time**(A) computes the cycle time of a (max,+) matrix A. Here, it is used for matrices of dimension at most $|\mathcal{P}|$. Then, the time complexity is $O(|\mathcal{P}|^3)$ using Karp's algorithm, see for instance [1].
– **Timing**(c) computes the sum of the firing times of the transitions along the circuit. The time complexity is linear.

In the algorithm below, the (max,+) representation of the behavior of live and 1-bounded free-choice nets is used. For every transition b, A_b is the (max,+) matrix representing the time behavior of the firing of b (see [7] for more details). The symbol $\otimes$ denotes the multiplication of matrices in the (max,+) algebra. This operation can be done in cubic time in the dimension of the matrices. Here, it is used for matrices of dimension at most $|\mathcal{P}|$.

Using the previous results, we get the following theorem.

Theorem 6.3. *Algorithm 3 finds the minimal throughput of a timed live and 1-bounded free-choice Petri net.*

Algorithm 3 : Worst-routing

Input: $(\mathcal{N}, M, \tau)$ a timed $(\tau \in (\mathbb{R}_+^*)^{\mathcal{T}})$ 1-bounded LBFC; and $\alpha = (\alpha_t)_{t\in\mathcal{T}} \neq (0,\ldots,0)$, a weight vector.
for all $b \in \mathcal{T}$ **do**
  $M_b \leftarrow$ **Blocking-Marking**(b);
for all $b, b' \in \mathcal{T}$ such that $\exists p,\ b \to p \to b'$ **do**
  $J \leftarrow$**Lightest-T-invariant**(b, b');
  $\sigma_{bb'} \leftarrow b, t_1, \cdots, t_m$ minimal firing sequence from M_b to $M_{b'}$ with transitions of J;
  $A_{bb'} \leftarrow A_b \otimes A_{t_1} \otimes \cdots \otimes A_{t_m}$;
  $\alpha_{bb'} \leftarrow \alpha_b + \alpha_{t_1} + \cdots + \alpha_{t_m}$;
Throughput $\leftarrow +\infty$;
Tmin $\leftarrow \emptyset$;
for all elementary circuit $c = t_1 \cdots t_k$ of $\mathcal{N}$ **do**
  $A \leftarrow A_{t_1t_2} \otimes A_{t_2t_3} \otimes \cdots \otimes A_{t_kt_1}$;
  $\alpha \leftarrow \alpha_{t_1t_2} + \cdots + \alpha_{t_kt_1}$;
  if cycle-time$(A) =$ **Timing**(c) **then**
    if $\alpha/$**cycle-time**(A) <Throughput **then**
      $J_{\min} \leftarrow \overrightarrow{\sigma_{t_1t_2}.\cdots.\sigma_{t_kt_1}}$;
      Throughput$\leftarrow \alpha/$**Timing**(c);

At each iteration in the first loop, the time complexity is $O(|\mathcal{T}|^3)$, and there are $|\mathcal{T}|$ iterations. Consider now the second loop. There are at most $|\mathcal{T}|^2$ iterations in the loop. At each iteration, we need to find a minimal sequence from M_b to $M_{b'}$ in a T-component. This can be done in time $O(|\mathcal{T}|^2)$. The length of a minimal sequence is of order $O(|\mathcal{T}|^2)$, see [8]. Hence the matrix $A_{bb'}$ can be computed in time $O(|\mathcal{P}|^3|\mathcal{T}|^2)$. At each iteration of the last loop, the time complexity is $O(|\mathcal{P}|^3|\mathcal{T}|))$ and there are as many iterations as elementary circuits in the net. Therefore, the total time complexity is $O(\mathcal{C}|\mathcal{P}|^3|\mathcal{T}|)$, where $\mathcal{C}$ is the number of elementary circuits. Since the number of elementary circuits can be exponential in the number of places ($O(2^{|\mathcal{T}|})$), the time complexity is exponential in the worst case. As for the space complexity, it remains polynomial in the size of the Petri net.

For comparison, consider the method of computation given (for rational timings) in Section 6.1 and which uses the *asap* marking graph. The size of the *asap* marking graph is exponential in the size of the original Petri net, more precisely its size is $O(2^{|\mathcal{P}|})$. So the complexity in time is $O((2^{|\mathcal{P}|})^3) = O(8^{|\mathcal{P}|})$ and the space complexity is at least $O(2^{|\mathcal{P}|})$. Observe that these complexities are evaluated without taking into account the necessity of transforming the Petri net with rational timings into an equivalent one with timings equal to 1, see Section 6.1. This transformation makes both time and space complexity of the classical method even worse.

Finally, remark that our construction also gives some insight on the period of the worse policy. Since the critical TT-invariant associated with the critical circuit (with n tokens) is the sum of n T-invariants, this means that the period of the worse routing policy in each routing place is periodic with a period $n \leq |\mathcal{P}|$. This is several order of magnitude smaller than the period that can be deduced from the classical algorithm which is exponential $O(2^{|\mathcal{P}|})$.

6.5 Bounded nets

If one considers a live and k-bounded free-choice net with $k \geq 2$, then the previous constructions for 1-bounded nets do not work anymore. If the timings are rational, the worst throughput is reached for periodic routings (Theorem 6.1), but the period is not bounded by the number of tokens in a circuit, as shown by the following example.

Example 6.4. Figure 6(a) represents a 2-bounded free-choice net where the period of the worst routing is greater that the number of tokens in any circuit. Furthermore the critical circuit for this routing is not elementary.

All the timings are equal to 1, as well as the weights. The routing in p_1 which gives the minimal throughput is, $(t1t4t4t4)^\infty$, and the periodic evolution is given in Figure 6(b).

The throughput of this evolution is 9/5, whereas if only the left (or right) event graph is live the throughput is 2. The period of the routing function of place p_1 is greater than the number of tokens in the circuits containing this place (2 tokens), as opposed to what happens in the case of 1-bounded nets. In this

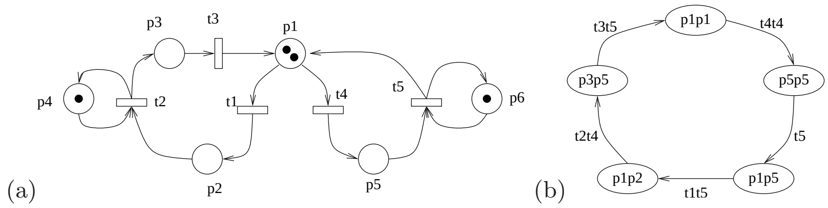

Fig. 6. Free-choice net and its worst evolution.

example, by changing the timings and/or increasing the number of transitions, but keeping the same number of tokens, it is possible to increase the period of the worst routing function arbitrarily. Therefore, bounding the period of the worst routing seems difficult, even for 2-bounded nets.

References

1. F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat. *Synchronization and Linearity.* John Wiley & Sons, New York, 1992.
2. A. Bouillard, B. Gaujal and J. Mairesse. *Extremal throughput in free-choice nets.* Research Report RR LIP 2005-14, ENS Lyon, France, 2005.
3. J. Carlier and P. Chretienne. Timed Petri net schedules. In *Advances in Petri Nets*, number 340 in LNCS, pages 62–84. Springer Verlag, 1988.
4. P. Chretienne. *Les Réseaux de Petri Temporisés.* PhD thesis, Université Paris VI, Paris, 1983.
5. G. Cohen, D. Dubois, J.P. Quadrat, and M. Viot. A linear system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Trans. Automatic Control*, 30:210–220, 1985.
6. J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Comp. Sc.* Cambridge Univ. Press, 1995.
7. S. Gaubert and J. Mairesse. Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE Trans. Aut. Cont.*, 44(4):683–698, 1999.
8. B. Gaujal, S. Haar and J. Mairesse. Blocking a transition in a free-choice net and what it tells about its throughput. *Journal of Computer and System Sciences*, 66(3):515-548, 2003.
9. J. Mairesse and L. Vuillon. Asymptotic behavior in a heap model with two pieces. *Theoret. Comput. Sci.*, 270:525–560, 2002.